Vol. 13. No. 01, Januari 2022: 25-28
http://publikasi.mercubuana.ac.id/index.php/jte
p-ISSN: 2086-9479 e-ISSN: 2621-8534

Benchmarking In Microcontroller Development Board Power Consumption For Low Power IoT WSN Application

Galang P. N. Hakim*, Muhammad Hafizd Ibnu Hajar, Ahmad Firdausi, Eko Ramadhan

Department of Electrical Engineering, Universitas Mercu Buana, Jakarta
*galang.persada@mercubuana.ac.id

***Abstrak*—One of the advantages of Wireless Sensor Network would be its ability to reduce cost of communication system using node to node communication. However Wireless Sensor Network also had a disadvantage which is has limited energy which is include this as low power application. This small energy capacity has limit WSN node capability to operate for a long time. In this paper, we compare power consumption for 3 popular microcontroller development platforms that use for fast development and prototyping Wireless Sensor Network node. The power consumption was including active mode (using most energy) and deep sleep mode (using least energy) operation. From benchmarking we can see that lolin ESP32 as a microcontroller development platform has the most efficient in power consumption which is only 40 mA in active and 0.05 in deep sleep mode, compare with arduino pro mini 8 mA in active and 0.3 mA in deep sleep mode, and wemos D1 mini 74 mA in active and 0.13 mA in deep sleep mode. This low power consumption in deep sleep mode has resulting in longer operational time which is almost 48 Month for lolin ESP32.**

***Keywords*—Low Power, Lolin ESP32, Wemos D1R2 Mini, Arduino Mini Pro, Deep Sleep.**

DOI: 10.22441/jte.2022.v13i1.005

I. INTRODUCTION

WSN (Wireless Sensor Network) was classified as a next generation network by ITU-T in 2010 [1]. It was define as a network that is comprised of interconnected sensor nodes that exchanging sensed data using wireless communication. Wireless sensor network is an interesting technology in sense of application and it cost to provide telecommunication for those applications. There are some advantages using wireless sensor network. First of all it can monitor its environment using sensors and sends its data monitoring using wireless communication [2]. Second its ability to reduce infrastructure deployment cost. And the third one it can extend coverage by using ad-hoc wireless communication methods [3].

However even though WSN has many advantages, it also has a disadvantage. A lot of WSN node can only store a small energy [4]. This happen because WSN node store power and using it to power up WSN node operation from battery [5]. While the energy that can store in battery was limited by its technology and mostly by its dimension [6]. This limited energy in battery has limit WSN node capability to operate for a long time [7].

Therefore to prolong WSN node operational time, some researchers are using additional power source (energy harvesting) such as soil energy harvesting [8], solar energy harvesting [9], and many other [10]–[12]. Some researcher try to use an efficient routing energy management such as LEACH [13], MUSTER [14], PDORP [15], and many others [16], [17]. In this paper we propose to use low energy management microcontroller feature such as deep sleep to prolong WSN nodal operational time. Going further we also compare 3 famous microcontroller development platforms and its power consumption in deep sleep mode against AA battery.

II. MATERIAL

In this paper we propose to compare 3 famous microcontroller development platforms and its power consumption in deep sleep mode. The reason why we are using these microcontroller development platforms is because of low cost, fast prototyping, and easy maintenance. For microcontroller development platforms we use would be:

A. Arduino Mini Pro

Arduino Mini Pro is one of microcontroller development platform that very popular to be use in IoT [18] or just for automation only [19]. There are two version of the Arduino Pro Mini, in order to achieve low power application we will chose that runs at 3.3V 8 MHz clock. However Arduino Pro Mini has voltage regulator on board so it can accept voltage up to 12V. The Arduino Pro Mini is a microcontroller development platform based on the ATmega328, therefore to calculate power consumption we need to see ATmega328 datasheet. ATmega328 has wide operating voltage from 1.8 V to 5.5 V with power consumption for active mode at about 200 uA and for deep sleep mode at about 0.1 uA [18]. In order to minimizing power consumption in ATmega328, ADC, Analog Comparator, Brown-out Detector, Internal Voltage Reference, Watchdog Timer, Port Pins, and On-chip Debug System feature need to be disable (please see ATmega328 architecture on figure 1) [20].

B. LOLIN D32

LOLIN D32 is one of microcontroller development platform that very popular to be use in IoT [21], [22]. LOLIN D32 is a low cost microcontroller development platform that based on Espressif official ESP32-WROOM-32 module. It has Wi-Fi & Bluetooth networking capabilities that essential as a lot of things today need to be connected to the internet (see figure 2). ESP32 integrates an antenna switch, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. ESP32-WROOM-32 module has wide operating voltage from 2.3 V to 3.6 V with power consumption for deep sleep mode at about 10 uA. In order to minimizing power consumption in ESP32-WROOM-32 module, The chip radio, CPU, and any other feature except RTC memory and RTC peripherals will be disable [23].

C. *Wemos D1R2 Mini*

Wemos D1R2 Mini (now LOLIN D1 Mini) is a popular low cost microcontroller development platform for IoT [24] that based on ESP8266EX. It has Wi-Fi networking capabilities that essential as a lot of things today need to be connected to the internet. ESP8266EX integrates antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules (see figure 3). ESP8266EX has wide operating voltage from 2.5 V to 3.6 V with power consumption for active mode at about 80 mA and for deep sleep mode at about 20 uA. In order to minimizing power consumption in ESP8266EX, The chip radio, CPU, and any other feature except RTC will be disable [25].

III. Method

In order to achieve low power application, we employ 3 methods. The first would be to use microcontroller deep sleep feature, the second one would be minimizing duty cycle of microcontroller active mode, and the third one would be battery complete charge cycle. The deep sleep mode feature already been discuss on the subsection before, therefore in this sub section we only discuss about duty cycle and battery charge consumption cycle only.

A. *Arduino Mini Pro*

In sensing application, sometimes we doesn't need sensor that always monitoring its environment. Such application was usually found in agriculture 4.0. In agriculture application it takes days to change environment variable value. One example in agriculture application would be basalt treatment. Basalt treatment is an activity to provide suitable environment so that plant can grow well. It was given to high acidity soil (pH ≤ 5.5) to increase its pH number. In halim et al research just to increase pH from 3.7 to 6.2 it would takes long time at about 56 days [27]. Therefore we can achieve low power WSN node that doesn't need every time monitoring, but instead we can just use one day one monitoring and the rest activity would be to sleep.

In order to achieve low power we need to minimize duty cycle. The duty cycle is defined as the fraction of time when the node is active in period of time, therefore duty cycle can be expressed as:

$$D = \frac{t_{active}}{T_{period}} \quad (1)$$

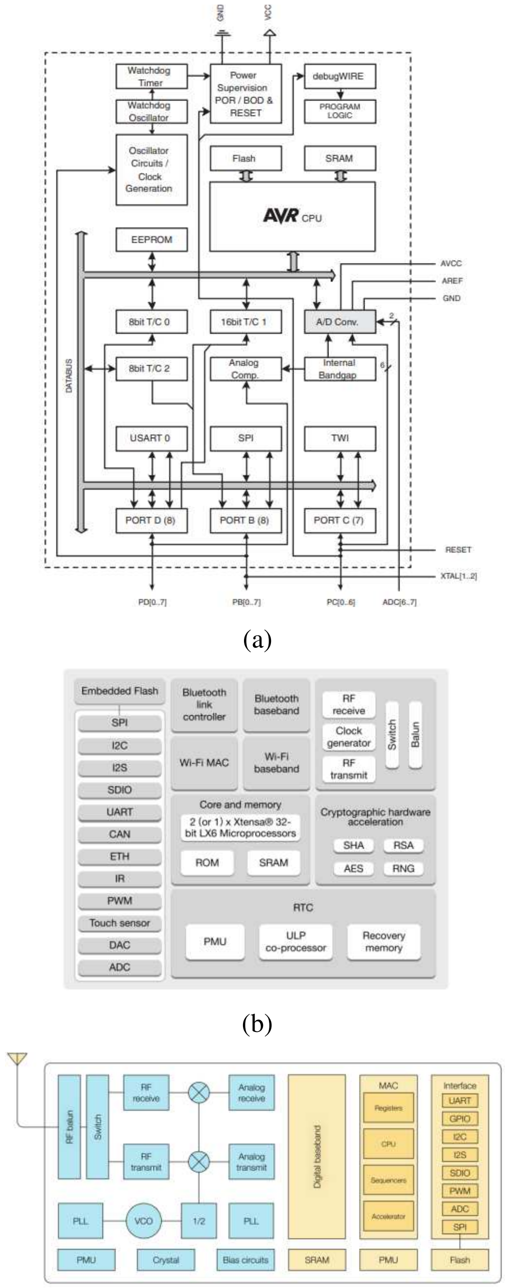
Where :

D : Duty Cycle

t_{active} : Time where WSN node active

T_{period} : Total period of time $t_{active}+t_{sleep}$

To minimize power consumption we intend to use 0.1% duty cycle for 1 day 1 monitoring scheme.



(a)

(b)

(c)

Figure 1. (a) Architecture of ATmega328 [18], (b) Architecture of ESP32 [26], (c) Architecture of ESP8266EX. [25].

B. *Battery Charge Consumption Cycle*

The battery charge consumption is composed number of sleep and active (measurement + transmission) charge consumption [28]. Therefore the charge consumption can be expressed by:

$$\mathrm{CC_{cycle}} = \mathrm{N_{sleep}\,CC_{sleep}\,t_{sleep}} + \mathrm{N_{active}\,CC_{active}\,t_{active}} \quad (2)$$

Where:

CC_{cycle} : Total charge consumption

N_{sleep} : Number of deep sleep in 1 period of time
CC_{sleep} : charge consumption on deep sleep mode
N_{sleep} : Number of active in 1 period of time
CC_{sleep} : charge consumption on active mode

The number of cycles that we can be performed with a battery capacity is given by:

$$N_{cycle} = \frac{Batt_{cap}}{CC_{cycle}} \quad (3)$$

Where:
N_{cycle} : Number of cycle can be performed
$Batt_{cap}$: Battery capacity

We can substitute equation 1 and 2 into equation 3, also in considering duty cycle 0.1% in one day period we can expressed

$$N_{cycle} = \frac{Batt_{cap}}{N_{sleep}\, CC_{sleep}\, 86313.6 + N_{active} CC_{active}\, 86.4} \quad (4)$$

IV. Results And Discussion

The first thing we do is measurement, and then we need to do comparison with datasheet manufacture. Table 1 shows charge consumption measurement versus chipset datasheet for each microcontroller development platform.

Table 1. Charge Consumption Measurement versus Datasheet for 1 second operational

Microcontroller Version	Charge Consumption Datasheet		Charge Consumption Measurement	
	Active (mA)	DeepSleep (mA)	Active (mA)	Deep Sleep (mA)
Arduino Pro Mini	0.2	0.0001	8	0.301
Wemos D1 Mini	80	0.02	74	0.13
LOLIN ESP32	68	0.01	40	0.05

In order for fair judgment we decide to not to activate wireless feature for wemos D1mini and lolin ESP32 because arduino mini pro doesn't have wireless capabilities. We can see that there is significant difference in charge consumption based on the measurement and datasheet for each chipset that use in the microcontroller development platform. We can see that this problem occur due to the regulator that use for each microcontroller development platform, also consume power (stay active) even when microcontroller chipset was on deep sleep mode.

To minimize power consumption for each microcontroller development platform for one day operation, deep sleep mode was executed with minimize duty cycle. Table 2 show measurement and calculation for 1 day operation using 0.1% duty cycle for each microcontroller development platform.

Table 2. Charge Consumption Measurement for 1 day operational

Microcontroller Version	Charge Consumption Measurement		Charge Consumption Measurement 1 day	
	Active (mA)	Deep Sleep (mA)	Active (mA)	DeepSleep (mA)
Arduino Pro Mini	8	0.301	691.2	25980.4
Wemos D1 Mini	74	0.13	6393.6	11220.8
LOLIN ESP32	40	0.05	3456	4315.7

Using equation 4 we can see that how many charge cycle for each microcontroller development platform last using energizer alkaline AA battery dimension (battery capacity 3000 mAh) [29]. Because we are using 1 day operation with 0.1 duty cycle we can also assume for each cycle was one day period. Table 3 shows how many cycle for each microcontroller development platform until battery depleted its energy storage (alkaline AA battery).

Table 3. Charge Consumption Measurement Perday Versus Battery Life For Microcontroller Platform

Microcontroller Version	Charge Consumption Each Mode perday		Battery Life in days (Energizer alkaline AA battery @3000 mAh)
	Active (mA)	Deep Sleep (mA)	
Arduino Pro Mini	691.2	25980.4	404.9
Wemos D1 Mini	6393.6	11220.8	613.1
LOLIN ESP32	3456	4315.7	1389.7

To make easy to understand figure 4 show benchmarking for each microcontroller development platform.

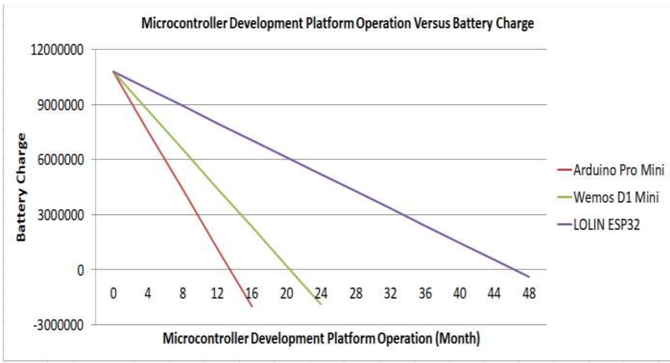


Figure 2. Benchmarking each microcontroller platform against operational days

We can see in figure 4 lolin ESP32 has longer operational time which is almost 48 month, while wemos D1 mini only 20 month, and Arduino pro mini is less than 16 month. With this we can say that lolin ESP32 as a microcontroller development platform has the most efficient in power consumption.

V. CONCLUSION

Microcontroller development platform was needed for fast prototyping WSN Node, unfortunately its power consumption is question whether it can support for low power application such as wireless sensor network. In this paper we benchmarking for efficient power consumption for each popular microcontroller development platform such as arduino pro mini, wemos D1mini, and lolin ESP32. From benchmarking we can see that lolin ESP32 as a microcontroller development platform has the most efficient in power consumption which is only 40 mA in active and 0.05 in deep sleep mode, compare with arduino pro mini 8 mA in active and 0.3 mA in deep sleep mode, and wemos D1 mini 74 mA in active and 0.13 mA in deep sleep mode. This low power consumption in deep sleep mode has resulting in longer operational time which is almost 48 Month for lolin ESP32. Even though ATMEGA328P was the lowest charge consumption in active 0.2 mA and in deep sleep mode 0.1uA, however when it became Microcontroller development platform, its charge consumption was rise gigantically which is active 8 mA and in deep sleep mode 0.301 mA that we assume this is because of their power regulator problem.

REFERENCE

[1] ITU-T, "Recommendation ITU-T Y.2221: Requirements for Support of Ubiquitous Sensor Network (USN) Applications and Services in the NGN Environment." p. 32, 2010, [Online]. Available: https://www.itu.int/rec/T-REC-Y.2221/en.

[2] N. Sharmin, A. Karmaker, W. L. Lambert, M. S. Alam, and M. S. T. S. A. Shawkat, "Minimizing the energy hole problem in wireless sensor networks: A wedge merging approach," *Sensors (Switzerland)*, 2020, doi: 10.3390/s20010277.

[3] J. Chen, S. Li, S. H. G. Chan, and J. He, "WIANI: Wireless infrastructure and Ad-hoc network integration," *IEEE International Conference on Communications*, 2005, doi: 10.1109/icc.2005.1495092.

[4] T. C. Hung and H. H. Trung, "Energy savings in applications for wireless sensor networks time critical requirements," *International Journal of Computer Networks and Communications*, vol. 8, no. 4, pp. 47–58, 2016, doi: 10.5121/ijcnc.2016.8403.

[5] M. T. Penella, J. Albesa, and M. Gasulla, "Powering wireless sensor nodes: Primary batteries versus energy harvesting," *2009 IEEE Intrumentation and Measurement Technology Conference, I2MTC 2009*, 2009, doi: 10.1109/IMTC.2009.5168715.

[6] Y. Ru, J. Kleissl, and S. Martinez, "Storage size determination for grid-connected photovoltaic systems," *IEEE Transactions on Sustainable Energy*, 2013, doi: 10.1109/TSTE.2012.2199339.

[7] X. Liu and J. Wu, "A method for energy balance and data transmission optimal routing in wireless sensor networks," *Sensors (Switzerland)*, vol. 19, no. 13, pp. 1–14, 2019, doi: 10.3390/s19133017.

[8] S. Budiyanto, G. P. N. Hakim, D. Septiyana, and A. Firdausi, "Soil Micro Energy Harvester for IoT-WSN Application," *Technology Reports of Kansai University*, vol. 62, no. 5, pp. 45–50, 2020.

[9] Y. Li, X. Li, and P. Wang, "A module harvesting wind and solar energy for wireless sensor node," *Communications in Computer and Information Science*, 2013, doi: 10.1007/978-3-642-36252-1_20.

[10] Mulyadi and R. W. Arsianti, "Low Power Electrical Generator from Soil Microbial Fuel Cell," *2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar, EECCIS 2018*, 2018, doi: 10.1109/EECCIS.2018.8692948.

[11] Y. Y. Choo and J. Dayou, "A Method to Harvest Electrical Energy from Living Plants," *Journal of Science and Technology*, 2013, doi: 10.1109/19.387319.

[12] I. Aramendia, U. Fernandez-Gamiz, E. Z. Guerrero, J. M. Lopez-Guede, and J. Sancho, "Power control optimization of an underwater piezoelectric energy harvester," *Applied Sciences (Switzerland)*, 2018, doi: 10.3390/app8030389.

[13] A. Wahab, F. A. Mustika, R. B. Bahaweres, D. Setiawan, and M. Alaydrus, "Energy efficiency and loss of transmission data on Wireless Sensor Network with obstacle," *Proceeding of 2016 10th International Conference on Telecommunication Systems Services and Applications, TSSA 2016: Special Issue in Radar Technology*, 2017, doi: 10.1109/TSSA.2016.7871084.

[14] L. Mottola and G. Pietro Picco, "MUSTER: Adaptive energy-aware multisink routing in wireless sensor networks," *IEEE Transactions on Mobile Computing*, 2011, doi: 10.1109/TMC.2010.250.

[15] G. S. Brar, S. Rani, V. Chopra, R. Malhotra, H. Song, and S. H. Ahmed, "Energy efficient direction-based PDORP routing protocol for WSN," *IEEE Access*, 2016, doi: 10.1109/ACCESS.2016.2576475.

[16] R. Bria, A. Wahab, and M. Alaydrus, "Energy Efficiency Analysis of TEEN Routing Protocol with Isolated Nodes," *Proceedings of 2019 4th International Conference on Informatics and Computing, ICIC 2019*, 2019, doi: 10.1109/ICIC47613.2019.8985668.

[17] G. Lukachan and M. A. Labrador, "SELAR: Scalable Energy-efficient Location Aided Routing protocol for wireless sensor networks," *Proceedings - Conference on Local Computer Networks, LCN*, pp. 694–695, 2004, doi: 10.1109/LCN.2004.111.

[18] P. S. Bhagat, D. V. S. Gulhane, and P. T. S. Rohankar, "Implementation of Internet of Things for Water Quality Monitoring," *International Journal of Trend in Scientific Research and Development*, 2019, doi: 10.31142/ijtsrd23655.

[19] S. Aryza, Z. Lubis, P. L. Eh Kan, W. Khairunizam, and Hamdani, "Enhanced of Speed Monitoring Brushless DC (BLDC) Equipment and Controller Based on Arduino," 2019, doi: 10.1088/1742-6596/1361/1/012049.

[20] Atmel, "ATmega328/P [Datasheet]," *AVR Microcontrollers*, 2016. .

[21] J. Fat and H. Candra, "Blockchain application in internet of things for securing transaction in ethereum TestNet," 2020, doi: 10.1088/1757-899X/1007/1/012194.

[22] S. Suraidi and S. Nathania, "Sistem Pengendali Smart-Kontak dengan Aplikasi Android dan Web," *TESLA: Jurnal Teknik Elektro*, 2020, doi: 10.24912/tesla.v21i2.7183.

[23] Espressif, "ESP32 Series Datasheet," *Espressif Systems*, 2019.

[24] M. Sabiran, D. Triyanto, and Suhardi, "Implementasi Wireless Sensor Network Pada Sistem Pemantauan Dan Pengontrolan Budidaya Tanaman Pada Rumah Kaca (Green House) Berbasis Website," *Jurnal Coding, Sistem Komputer Untan*, 2018.

[25] Espressif Systems, "Datasheet ESP8266EX," 2020.

[26] Espressif Systems, "Datasheet ESP32 Series," 2019. [Online]. Available: www.espressif.com.

[27] N. S. adah Abdul Halim, R. Abdullah, S. A. Karsani, N. Osman, Q. A. Panhwar, and C. F. Ishak, "Influence of soil amendments on the growth and yield of rice in acidic soil," *Agronomy*, 2018, doi: 10.3390/agronomy8090165.

[28] C. J. García-Orellana, M. Macías-Macías, H. M. González-Velasco, A. García-Manso, and R. Gallardo-Caballero, "Low-power and low-cost environmental iot electronic nose using initial action period measurements," *Sensors (Switzerland)*, 2019, doi: 10.3390/s19143183.

[29] Energizer, "Product Datasheet Energizer E91 Form No. E91NA1018," 2021.